

回帰木を用いた非線形写像の学習と識別問題への応用

和田 俊和 中村 恭之

和歌山大学 システム工学部 情報通信システム学科

あらまし 本報告では、機械学習およびデータマイニングの分野で主にスカラ関数の近似に用いられてきた回帰木 (Regression Tree), およびそれを線形回帰に拡張した線形回帰木 (Linear Regression Tree) の研究を概観し、筆者らが新たに開発した区分的線形写像木 PaLM-tree について述べる。PaLM-tree は、線形回帰木の出力次元を多次元に拡張したものであり、木の構築時に Split-and-Merge 戦略を採用することにより、学習時間の短縮と写像の高精度化を同時に達成している。また、写像計算は二分探索と線形変換だけであるので、極めて高速である。PaLM-tree の用途は非常に多岐に渡るが、本報告では画像の領域分割、カメラキャリブレーション、そして任意の識別器の機能を模倣する問題への応用例について述べる。

Regression Trees and Its Application to Classification Problems

Toshikazu Wada and Takayuki Nakamura

Department of Computer and Communication Sciences

Faculty of Systems Engineering, Wakayama University

Abstract This paper presents evolutionary history of regression trees, a novel regression tree called PaLM-tree, and its application to Computer Vision and Pattern Recognition problems. Regression tree is a tree representation of nonlinear function, which is mainly investigated in the fields of Machine Learning and Data Mining. The original regression tree is a simple tree representation holding function values at leaves (terminated nodes). Linear regression tree extends the function values to linear regression coefficients. Since these regression trees are designed for rule-extraction, most of them are single valued for representing nonlinear scalar functions. PaLM-tree is a multi-valued extension of the linear regression tree with Split-and-Merge tree construction. PaLM-tree can be applied to varieties of nonlinear mapping learning problems. This paper also presents applications of PaLM-tree involving image segmentation, camera calibration and imitating classifiers.

1. はじめに

コンピュータビジョン、特に Physics based Vision では、事象と事象の関係を物理則に基づく因果関係としてとらえ、それを数理モデルで表現する研究が数多く行われてきた。しかし、このような方法で現実の様々な事象を記述しようとする時、

- 法則やルールが複雑かつ膨大になってしまう。
- そもそも、因果関係を解き明かすことが非常に困難な問題が多数存在する。

などの問題に直面し、結局、このようなアプローチだけでは、現実世界で動作する画像認識システムを構築す

ることは極めて困難であると言える。

このように、事象間の因果関係が不明であり、具体的な事例から相関関係しか求めることができない問題は数多く存在する。この種の問題に対しては、因果関係を数理モデルとして表現せず、「この場合には、こうなる」という現実に起きた事例間の相関関係のみを学習するだけで、ある程度単純な視覚認識機能を実現することができると考えられる。

このような相関関係、すなわち非線形写像、を学習する問題は、画像情報からその解釈への写像を求める問題、あるいは、連続特徴ベクトルから離散クラスラベルへの写像を求めるパターン認識の問題など、数多くの場面で必要とされている根本的技術である。

これまでに提案されてきた非線形写像の学習法は、以下の3つに大別される。

〒640-8510和歌山県和歌山市栄谷930 和歌山大学システム工学部情報通信システム学科

1. ニューラルネットワーク(NN)法[1], Group Method of DataHandling(GMDH)法[2], RBFネットワーク(RBFnet)法[3] や NGnet 法[4]など, 写像全体を連続な合成関数で近似する手法
2. 事例として与えられた入力と出力のペアをそのまま記憶しておく k 近傍(k-NN)法[5],[6]
3. 入力空間を階層的に分割して木構造で表し, その葉の部分に写像先の値を格納しておく回帰木 (Regression Tree)[7]や, それを線形回帰に拡張した線形回帰木 (Linear Regression Tree)[8]

これらには,それぞれ長所や短所があるものの,最も潜在的な能力が高いものは回帰木であると我々は考えている. その理由は以下の通りである.

1. NN,GMDH,RBFnet,NGnet 法では,
 - 連続な写像関数が得られるため, ビジュアルサーボなど微分計算を伴う解析に適する.

という長所がある一方で,

- ネットワークの構造を決定するためのパラメータを予め設定しなければならず¹, 複雑な写像を学習する場合, ネットワークの初期重み係数値を適切に設定しないと学習が収束しない.
- 個々のデータに対する写像の推定精度が保障できない.

という欠点がある.

2. k-NN 法では,
 - 入出力関係を事例の集合のみで表し, 複雑なモデルを用いていないため, 事例を記憶するだけで特に学習は必要ない.

という長所がある一方で,

- 入力次元数が大きい場合, 複数の類似事例を検索する k 最近傍探索処理に多くの時間を費やし写像計算が高速に行えない.
- 事例データをそのまま記憶するため, 大量のメモリを消費してしまう.

などの欠点がある.

1. これらに対して回帰木は,
 - 原理的に任意の精度で, 関数近似を行うこ

¹ NN法では,レイヤー数, 中間素子数, 初期重み係数など, GMDH 法では, レイヤー数, 基底関数, 変数選択の方法, RBFnet 法では, 中間素子数やガウス関数の形状を規定するパラメータ, ネットワークの重み計算用パラメータ, NGnet 法では, ガウス関数の数, ガウス関数の形状を規定するパラメータや数種類の学習パラメータを指す.

とができ, 学習は必ず収束する.

- 基本的に関数近似精度を与えるだけで学習が行え, 複雑なパラメータの初期値調整の問題はない.
- 近似計算を行う際には, 二分探索木による探索と, 簡単な計算しか必要としないため, 近似計算は極めて高速である.
- 得られた木構造は IF-THEN ルールに変換することができ, ルールや知識発見などの用途に利用できる.

などの利点がある一方で,

- 連続な写像関数が得られない.

という欠点がある. しかし, この欠点は α ブレンディングなどの平滑化処理を行うことで簡単に解消することができる. このように, 回帰木には克服困難な著しい欠点は見当たらない.

本稿では, この回帰木を中心としたこれまでの研究の流れを概観し, それを一般化しコンピュータビジョンやパターン認識の問題に利用可能なレベルにまで引き上げた PaLM-tree アルゴリズムについて説明する.

2. 非線形写像学習問題

非線形写像の学習問題とは, p 次元入力ベクトル

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T, \quad (1 \leq i \leq n)$$

に対して q 次元出力ベクトル $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{iq})^T$ が得られて

いるとき, 入出力関係 $\mathbf{y}(\mathbf{x})$ を推定する問題である.

この推定にパラメータ $\boldsymbol{\theta}$ によって決まる $\mathbf{Y}(\mathbf{x}; \boldsymbol{\theta})$ なるモデルを用いる場合には,

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \|\mathbf{y}(\mathbf{x}) - \mathbf{Y}(\mathbf{x}; \boldsymbol{\theta})\|$$

を求めるとなる. ところが, $\mathbf{Y}(\mathbf{x}; \boldsymbol{\theta})$ が事前に与えられた固定的なモデルである場合, データとして与えられた \mathbf{x}_i と \mathbf{y}_i の関係をうまく表現できるか否かは保証できない. 仮に, 十分に複雑な写像を表現できるモデルを用いた場合には, $\boldsymbol{\theta}$ の次元数が極めて高くなり, 上述のパラメータ推定計算が不安定になってしまうため, このような写像学習の手法には潜在的な限界がある.

ここで, 単純な関数近似モデルを用い, \mathbf{x}_i が属す範囲ごとに近似モデルを切り替える方法について考えてみる. これは, 入力空間 D を直和分解によって,

$$D = \bigcup_{j=1}^M D_j \quad \text{但し, } D_k \cap D_j = \begin{cases} \phi & j \neq k \\ D_j & j = k \end{cases}$$

としたとき、 D_j 毎に θ を切り替える手法である。この場合、各 D_j に含まれる \mathbf{x}_i の個数を少なくしていけば $Y(\mathbf{x}_i; \theta)$ と y_i の差を小さくすることができる。このような考えに基づく手法が回帰木である。

3. 回帰木

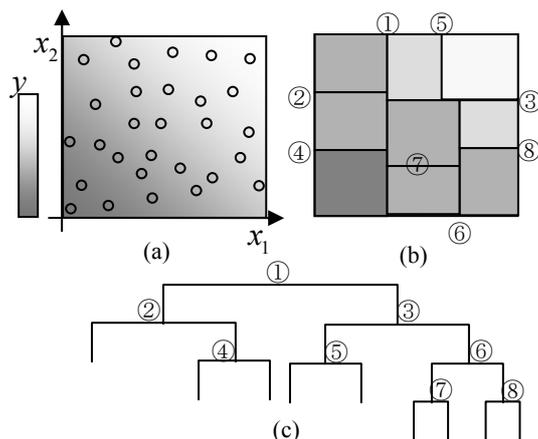


図1 回帰木

回帰木は、Breiman[7]らによって提案された関数近似の手法であり、現在もお機械学習やデータマイニングの分野で研究が進められており、基本的なツールとしても使用されている。

図1は2次元の入力から1次元出力への写像を求めた問題を例とした回帰木の問題図である。同図(a)の学習データは位置 $\mathbf{x} = (x_1, x_2)^T$ が入力であり、その点での濃度が出力 y を表している²。このとき、回帰木は同図(b)のようになる。すなわち、入力空間を矩形領域に分割し、個々の領域内の近似値を単一の値として保持しておくというものである。入力空間の分割は、同図(c)のような木構造で表現することができ、これを二分探索木として用いることにより、近似計算を行う際に入力されたデータがどの矩形領域内に入ってきたのかを高速に探索できる。先に述べた矩形領域内の近似値は、この二分探索木の葉の部分に格納される。

このような回帰木を構築する際の問題は、

- 1) 何を基準として分割判定を行い、
- 2) どこで空間を分割するのか、

ということである。初期の研究では、1)の問題に関して

² 実際には与えられる学習データは稠密ではなく、離散的な点のデータとなる。

は、各矩形領域内の出力 y の分散値があらかじめ与えておいた閾値を越えるか否かによって分割を行うかどうか判定されていた。2)の問題に関しては、最も分散が大きくなる軸を選択し、分割によって生成される2つの矩形領域内の分散の和が最小となる位置で分割が行われていた。

しかし、回帰木の葉に出力値を埋め込むという方法だけでは、木のサイズが大きくなりすぎ、また近似も階段状になってしまうという問題が残っていた。この問題を解決するため Quinlan[8]は各矩形領域に線形回帰式をあてはめる線形回帰木を提案した。線形回帰木では、定数の出力値を用いる場合よりも回帰の当てはめがうまくいくため、一般に矩形領域のサイズは大きくなり、木の高さも平均的に低くなるため、二分探索の計算も高速化する傾向がある。唯一の速度低下要因は、葉ノードに格納される回帰係数と入力データとの乗算に若干の時間がかかるだけである。

但し、Quinlan の方式においても空間の分割判定および分割位置の決定に矩形内での分散が用いられていたため、その後の研究では以下のようにさまざまな分割方式が提案されている。

Karalic[11]によって提唱された分割方式は、線形回帰式をあてはめ、平均回帰誤差が指定した値以下になるまで分割を繰り返し、分割位置も線形回帰が最もうまくいくように決定する方法である。この方式は分割位置を決定する際に何度も線形回帰式の当てはめ計算を行うため、計算時間が長いという欠点がある。Alexander[12]らは回帰の入力を一次元に固定した場合の効率的な計算法を提案している。Chaudhari[13]は定数、線形、そして高次の多項式の当てはめを全ての節で行う方式を提案しており、その中で分割位置を回帰残差の符号を用いて決定する方式を提案している。Dobra[14]らは EM アルゴリズムで2つの正規分布を出力に当てはめることによって、分割位置を決定している。このように、さまざまな分割判定や分割位置の決定方式が提案されているが、未だに決め手となる方式は存在しないと言える。

これまでの研究を概観すると、

- 回帰木の計算ではデータからのルール発見を主な目的としていたため、一般にルールが複雑化する多次元ベクトル空間を値域とする拡張は検討されてこなかった。
- 入力空間の分割方法が多様であり、簡便且つ信頼性の高い手法が存在しない。

などの問題点が浮上してくる。このため、コンピュータビジョンやパターン認識といった用途に本格的に用いるこ

とができるまでには成熟していないと言える。

4. PaLM-tree

4.1. 概要

前章で述べた回帰木の問題点を解消するため、我々は次の2点を拡張した回帰木を提案した。

- 出力ベクトルを多次元に拡張し、 p 次元から q 次元への写像一般を取り扱えるようにする。
- 分割位置の決定は厳密には行わず、Split-and-Merge戦略を用いる。

これらの拡張を施した回帰木を区分線形写像木 (Partially Linear Mapping tree: 以下 PaLM-tree) と呼ぶ。

出力次元を多次元ベクトルとすること自体に技術的な困難さはないが、この拡張により応用範囲がかなり広がる。たとえば、入力空間に出力を多様体として埋め込むパラメトリック固有空間法で扱えるのは $p \geq q$ の場合に限られていたが、PaLM-tree では、 $p < q$ の場合も扱えるため、 $(x, y) \mapsto (r, g, b)$ のように位置から色への写像を求める画像の領域分割の問題などにもそのまま適用できる。

分割位置決定の問題に対しては、Horowitz と Pavlidis[9]の Split-and-Merge による画像の領域分割法の根底にある考え方、すなわち分割位置を厳密に決定するのではなく、とりあえず条件を満足するまで分割を行い、後から条件を満足する範囲で領域をまとめれば、条件を満足する最大の領域が得られるという考え方を採用している。

詳しく述べると、分割軸は常に矩形の最長軸を選択し、分割によって分割後の矩形内のデータ数に偏りが出ないように分割位置を決定している。このような方式を用いる主な理由は計算速度を重視しているためである。この分割によって得られた領域内のデータに対して回帰計算を行い、回帰誤差が指定された範囲内に収まるまで分割を再帰的に行っていく

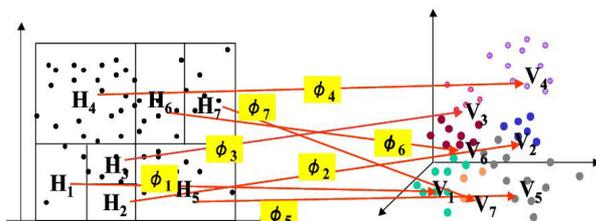


図2 PaLM-tree の概念図

この分割が終了した後に、隣接する矩形内のデータをまとめ、共通の回帰係数を持たせること、すなわち

Merge ができるか否かを判定するために回帰計算を行う。この計算では Merge によって新たな矩形ができずとも回帰係数を共有化できるようにしている。

分割が進んでいくと矩形内のデータ数が減少していく。このとき、データ数が入力空間の次元数 p 以下になると回帰計算が行えなくなるため、一般逆行列を用いた回帰計算を行っている。

4.2. 回帰計算

まず、以下の説明で使用する変数の定義を与える。

入力ベクトル $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T$ に対応する出力ベクトルを $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{iq})^T$ とする。これらのデータ組が n 個得られたとき、全データは、

$$X = [\mathbf{x}_1 \ \dots \ \mathbf{x}_n] \quad Y = [\mathbf{y}_1 \ \dots \ \mathbf{y}_n]$$

という2つの行列で表現することができる。

B を $p \times q$ の係数行列としたとき、 \mathbf{x}_i に対する回帰式は $B(\mathbf{x}_i - \bar{\mathbf{x}}) + \bar{\mathbf{y}}$ となる。但し、 $\bar{\mathbf{x}}$ 、 $\bar{\mathbf{y}}$ はそれぞれ X, Y の平均ベクトルである。ここで、

$$X' = [\mathbf{x}_1 - \bar{\mathbf{x}} \ \dots \ \mathbf{x}_n - \bar{\mathbf{x}}] \quad Y' = [\mathbf{y}_1 - \bar{\mathbf{y}} \ \dots \ \mathbf{y}_n - \bar{\mathbf{y}}]$$

とし、誤差ベクトル $\mathbf{e}_i = \mathbf{y}_i - B\mathbf{x}_i$ を導入することによって、 $Y' = BX' + E$ という式が得られる。但し、 $E = [\mathbf{e}_1 \ \dots \ \mathbf{e}_n]$ である。

この式に SVD によって得られた X' の一般逆 X'^{\dagger} を右側からかけると $B = Y'X'^{\dagger} - EX'^{\dagger}$ という式が得られるため、 $B = Y'X'^{\dagger}$ と回帰係数を推定できる。この回帰係数の計算法は主成分回帰分析と呼ばれる方法であり、独立なデータ数が入力空間の次元数 p 以下でも破綻することなく実行することができるが知られている。

4.3. 回帰誤差の評価

ここでは、PaLM-tree において矩形領域を分割するか否かを判定する際の誤差評価の方式について説明する。しかし、この評価は適用する問題に依存して決定すべき内容であり、以下に述べる方式が唯一ではないことに注意されたい。

単純に矩形領域内の回帰誤差の平均値 $\frac{1}{n_i} \sum_i \|\mathbf{y}_i - \mathbf{y}(\mathbf{x}_i)\|$ を用いた場合には、次に示すような不都合が生じる。 \mathbf{y}_i は q 次元ベクトルであり、一つの矩形領域内部で、ある成分は変動が大きく、ある

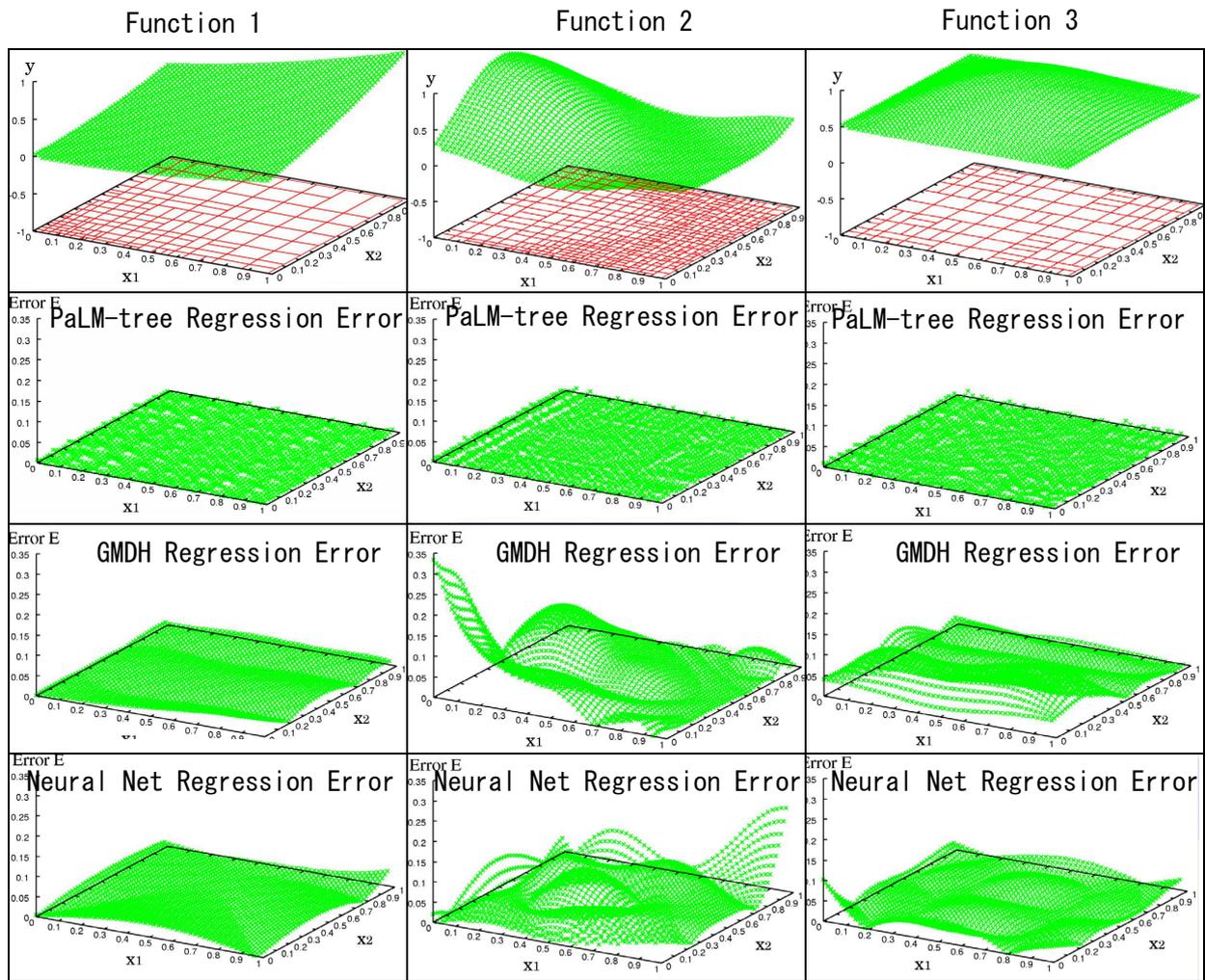


図3 シミュレーション結果

成分は変動が小さいということが一般的に起こりうる。このため、変動の大きさを考慮した誤差評価を行わなければならない。 $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{iq})^T$ としたとき、

y_{ij} の偏差を σ_j と表すものとする。このとき、

$$e_j = \sum_i \frac{\|y_{ij} - y_j(\mathbf{x}_i)\|}{\sigma_j} \text{ とし、 } e = \sum_j e_j \text{ によって}$$

回帰誤差を評価する。これによって、各成分の変動幅の影響を受けにくい分割判定が行える。

4.4. シミュレーション

ここでは、従来法として GMDH 法, NN 法を取り上げ、以下に示す 3 種類の関数の近似について提案手法との性能比較を行った結果を示す。なお、各入力変数 x_i

や係数 c_i については区間 $[0, 1]$ 内で発生した乱数を用いて決定しており、入力次元数 p や出力次元数 q 、およびデータ数も変化させた実験を行っている。

Function1 $y_q = (c_1 x_1 + c_2 x_2 + \dots + c_p x_p)^2$

Function2 $y_q = (c_1 \cos(\pi x_1) + c_2 \sin(\pi x_2) + \dots + c_{p-1} \cos(\pi x_{p-1}) + c_p \sin(\pi x_p))^2$

Function3 $y_q = 0.5 + 0.5 \prod_{i=1}^p c_i \sin(\pi x_i)$

GMDH は、大谷らによって提案された NF-GMDH を用いており、Neural Net は 3 層のネットワークを誤差逆伝播でトレーニングしたものを用いている。これらの手法はいずれも、出力の偏差に対して 1% 以内の誤差にな

るようにトレーニングを行っている。PaLM-tree ではすべてのケースで指定した誤差範囲内に収束しているが、GMDH, Neural Net では収束しないケースもあった。

まず、一例として $p = 2, q = 1, c_i = 1$ としたときの回帰結果を図 3 に示す。この図では最上段が近似の対象となる関数、2 段目以降がそれぞれ PaLM-tree, GMDH, Neural Net での近似誤差となっている。最上段のグラフには、PaLM-tree によって入力空間がどのように分割されたかを $y = -1$ の平面上に図示している。この図を見ても明らかのように、PaLM-tree では回帰誤差が非常に小さくなっていることが分かる。これに対して、GMDH, Neural Net では Function 2, 3 に対して大きな誤差が出ていることが分かる。事実これらの結果では GMDH, Neural Net は指定した 1% の誤差範囲内に収束していない。

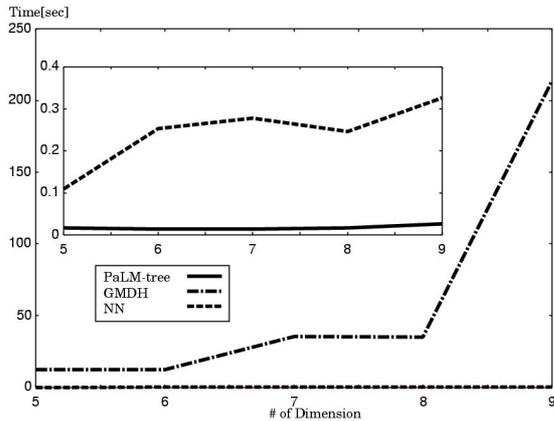


図 4. 次元数の変化に対する画像学習時間の変化

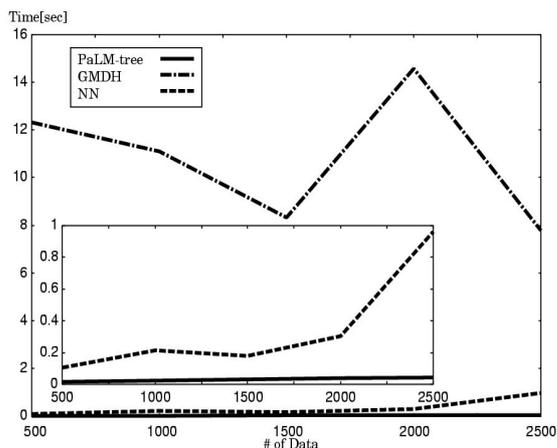


図 5. データ数の変化に対する画像学習時間の変化

画像の学習時間の比較を図 4, 5 に、入力データが与えられたときに画像を計算する時間の比較を図 6, 7 にそれぞれ示す。画像の学習時間については、次元数の

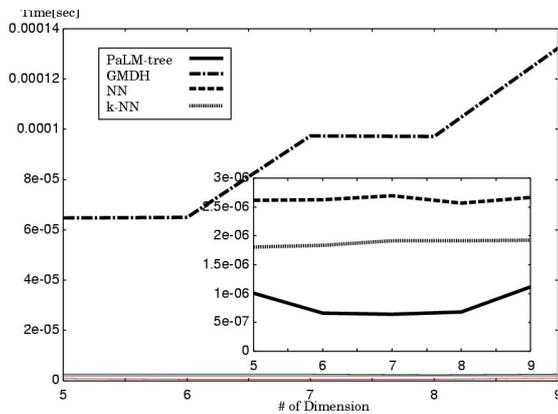


図 6. 次元数の変化に対する画像計算時間の変化

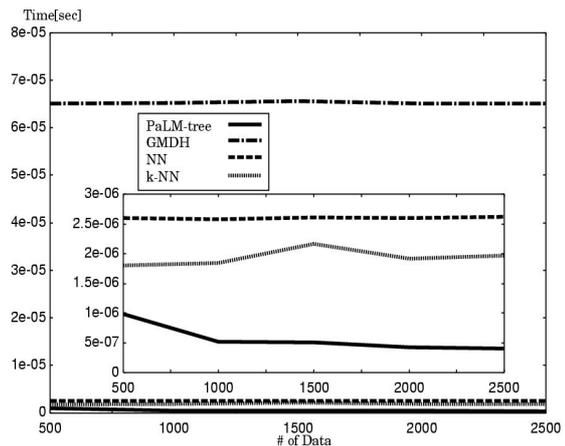


図 7. データ数の変化に対する画像計算時間の変化

変化に対する特性と、学習データ数に対する特性を PaLM-tree, GMDH, Neural Net の 3 つについて示している。この図からわかるように、PaLM-tree は他の手法と比べて短時間で学習が終わることが分かり、次元数やデータ数の変化に対して非常にゆっくりとしか学習時間が変化しないことがわかる。また、図 6, 7 に示した画像計算時間では、PaLM-tree, GMDH, Neural Net, k-NN 法の比較結果を示している。この図からわかるように、画像計算においても PaLM-tree が極めて優れていると言える。

5. PaLM-tree の応用例

PaLM-tree の用途は非常に多いが、ここでは典型

的な応用例3つを示す。

5.1. 画像の領域分割 ($R^2 \mapsto R^3$)

画像は位置 (x, y) から色 (r, g, b) への写像として捉えることができる。すなわち、 $R^2 \mapsto R^3$ という写像である。これを学習することは、画像の領域分割を行うことに他ならない。図 8 に示した例は、グラデーションをつけた複数の領域から成る画像データに対して領域分割を行った場合の結果である。

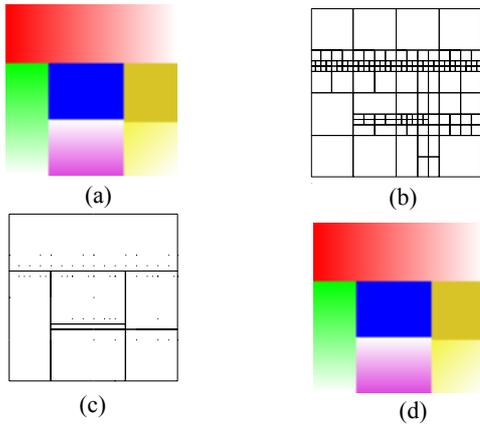


図 8. 画像の領域分割結果

図 8(a)は原画像、(b)は分割のみを行った場合の矩形領域であり、(c)は引き続き統合処理を行った結果である。(d)は最終的に得られた回帰係数から画像を復元した結果である。(b)の領域数は 143 個であり、(c)では 7 個まで減少している。このことから、統合計算の有効性が確認できる。

5.2. カメラキャリブレーション ($R^2 \mapsto R^2$)

歪のあるカメラによって得られた画像を補正する問題は、位置 (x, y) から補正後の位置 (x, y) への写像としてとらえることができる。ここでは、自由曲面ミラーを用いた全方位カメラ(図 9(a))のカメラ装着位置をずらすことによって歪みを生じさせた画像のキャリブレーションを行った結果を示す。図 9(b)にその撮影風景を示す。

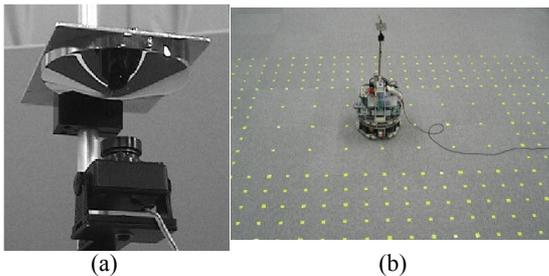


図9. 自由曲面ミラーを用いた平面領域の撮影

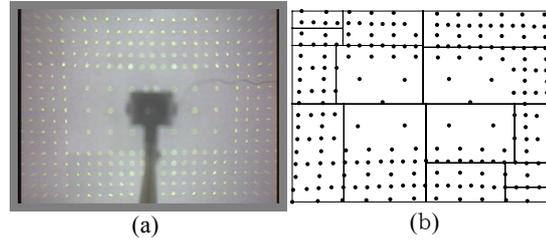


図 10. 歪んだ画像とそのキャリブレーション結果

このとき撮影された画像を図 10 に示す。被写体は平面上に規則的に配置された点群であるが、同図(a)に示す画像には歪みが生じている。画像上の点群の位置を入力ベクトル、平面上の点群の位置を出力ベクトルとして PaLM-tree で学習を行った結果を同図(b)に示す。この図から、歪んだ点配置が正射影に近い規則的配置に戻されていることが確認できる。

5.3. 識別器の模倣 ($R^d \mapsto R^c$)

通常のパターン識別機構は、 d 次元ベクトルからクラス数に相当する c 次元ベクトルに写像する機構としてとらえられる。これまでも様々な識別器が提案されてきたが、SVM, Adaboost など識別能力が高いものは、必ずしも高速な識別器として動作する保証はない。特に複数の識別機構を組み合わせるアンサンブル学習では、識別時に高速な計算を行わせることが困難である。このような場合、PaLM-tree を用いて識別機構の入出力関係を学習しておけば、認識能力は高いが低速な識別器を高速化することができる。

以下に示す実験では、識別器として k -最近傍識別器を用いて、その機能を学習させている。クラス i の出力は $-1 \leq y_i \leq 1$ という範囲に正規化されており、 k 個の最近傍プロトタイプのうち、クラス i のクラスラベルを持つものを $+1$ 、それ以外を -1 としてカウントし、その総和 s_i を k で除したものを出力ベクトルの要素としている。

すなわち、 $\mathbf{y}(\mathbf{x}) = \frac{1}{k}(s_1, s_2, \dots, s_c)^T$ である。この

ベクトルの要素のうち、最も値が大きいものが s_r であったとすると、入力 \mathbf{x} はクラス r に分類される。

以下に示す例では、偏差 σ が互いに等しい等方性正規分布に従う乱数によって生成した二次元データを 2 クラスに分類する k -最近傍識別器の出力を模倣した結果を示している。 $k = 20$ であり、トレーニングデータは各 500 個、合計 1000 個のデータである。参考のため、線形 SVM を用いて識別を行った結果も示している。正規分布に従う乱数の分布は互いに重なりを持っており、

	1σ	2σ	3σ	4σ
k-NN	69.20%	84.80%	93.00%	97.90%
PaLM	69.00%	84.25%	92.80%	97.85%
SVM	64.3%	75.0%	85.7%	91.1%
BayesErr	30.1%	16.1%	6.8%	2.3%

表 1. 2つの正規分布の識別結果

分布間の距離が小さくなると、重なり部分が多く、大きくなると重なりが少なくなる。表1ではこの距離を σ の倍数で表している。このような重なり部分では識別の誤り確率を0にすることはできない、この誤り確率を Bayes 誤り確率と呼ぶ。表には数値的に計算した Bayes 誤り確率も示されている。

この表を見る限り、PaLM-tree は k-最近傍識別器の出力を多少の誤差はあるもののうまく模倣できていることが分かる。

6. まとめ

非線形写像学習の方法として、回帰木の概要とそれを一般化、高速化した PaLM-tree について述べた。実験結果からも、GMDH, Neural Net, k-NN 法に対する PaLM-tree の優位性は明白であると言える。また、PaLM-tree の用途はコンピュータビジョン、制御、計測、パターン認識など非常に多岐にわたり、これからも応用が進められていくものと期待される。

回帰木そのものに関しては、木の分割方法と後処理だけでなく、回帰を行う関数系を高次にするもの[13]やカーネル関数を導入して個々の回帰の性能を向上させるもの[15]などの研究が進められており、今後さらなる発展が期待できる。

謝辞 5.3 節の実験を行ってくれた、和歌山大学システム工学部情報通信システム4回生林拓君に感謝いたします。本研究の一部は文部科学省科学研究費補助金基盤研究(A)(2)16200014の補助を受けている。

参考文献

[1] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning representations by back-propagation errors," *Nature* 323(1986), 533-536.

[2] A. G. Ivakhnenko, "Polynomial theory of complex systems," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 364-378, 1971.

[3] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, 78(9),

pp.1481-1497, September 1990.

[4] L. Xu, M. I. Jordan, and G. E. Hinton.: "An alternative model for mixtures of experts," In *Advances in Neural Information Processing Systems 7*, pp. 633-640, The MIT Press, 1995.

[5] R. O. Duda and P.E. Hart, "Pattern Classification and Scene Analysis," *New York, Wiley*, 1973.

[6] C. G. Atkeson, A. W. Moore, and S. Schaal: "Locally Weighted Learning," *Artificial Intelligence Review* 11, pp.11-73, 1997.

[7] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone: "Classification and Regression Trees," *Chapman & Hall, New York*, 1984.

[8] J. R. Quinlan: "Learning with continuous classes," In *Proc. of 5th Australian Joint Conference on Artificial Intelligence*. pp. 343-348, 1992.

[9] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a tree traversal algorithm," *Journal of The Association for Computing Machinery*, Vol. 23, No.2, pp.:368-388, April 1976.

[10] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. "An optimal algorithm for approximate nearest neighbor searching," In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pp. 573-582, 1994.

[11] A. Karalic, "Linear regression in regression tree leaves," In *International School for Synthesis of Expert Knowledge*, Bled, Slovenia, 1992.

[12] W. P. Alexander and S. D. Grimshaw, "Treed regression," *Journal of Computational and Graphical Statistics*, No.5, pp. 156-175, 1996.

[13] P. Chaudhuri, M.-C. Huang, W.-Y. Loh, and R. Yao, "Piecewise-polynomial regression trees. *Statistica Sinica*," 4, pp.143-167, 1994.

[14] A. Dobra and J. E. Gehrke, "SECRET: A Scalable Linear Regression Tree Algorithm," In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Edmonton, Alberta, Canada, July 2002.

[15] L. Torgo. Functional models for regression tree leaves. In *Proc. 14th International Conference on Machine Learning*, pages 385-393. Morgan Kaufmann, 1997

[16] 大谷 崇, 市橋秀友, 長坂一徳, 三好哲也: "誤差逆伝播学習によるニューロ・ファジィ GMDH での関数近似-階層型ニューラルネットワークとの近似精度の比較," *日本経営工学会論文誌*, Vol.47, pp.384-392, 1997.